

SMARTBRIDGE Virtual Internship Program in collaboration with APSCHE

PROJECT REPORT

Project Title:

HematoVision: Advanced Blood Cell Classification Using Transfer Learning

Submitted By:

Name: Bura Naga Rohith

Team ID: LTVIP2025TMID38768

Internship Domain: Artificial Intelligence & Machine Learning (AI-ML)

Organization: SmartBridge in collaboration with APSCHE

Submitted To:

Industry: SmartBridge

Submission Date: 29th June 2025

1. INTRODUCTION

In recent years, artificial intelligence (AI) and machine learning (ML) have transformed the medical domain by enabling faster and more accurate diagnosis of various diseases. One such critical area in healthcare is the analysis of blood cells, which plays a vital role in identifying infections, anemia, leukemia, and other hematological disorders. Traditionally, examining blood cells required manual inspection by medical experts using microscopes, which is time-consuming and prone to human error.

To address this challenge, our project **HematoVision_AI** aims to automate the classification of blood cell images using deep learning models. The project focuses on building a Convolutional Neural Network (CNN)-based model that can accurately classify different types of blood cells. Additionally, the model is integrated into a user-friendly Flask-based web application that allows users to upload blood cell images and receive instant predictions.

This report outlines the complete journey of developing the HematoVision_AI system—from understanding the dataset, training the model, to building the frontend and backend for real-time predictions.

2. Problem Statement and Objectives

Problem Statement

Manual identification and classification of blood cells under a microscope are time-consuming, require expert knowledge, and are prone to human error. This method is not scalable in regions with limited medical expertise and resources. Automating this process using deep learning can support pathologists by providing faster and more reliable results for early diagnosis and treatment planning.

Objectives

The primary objectives of the HematoVision_AI project are:

1. **To develop a machine learning model** capable of classifying blood cell images accurately using deep learning techniques such as Convolutional Neural Networks (CNNs).
2. **To integrate the trained model** into a web-based platform using Flask, enabling users to upload blood cell images for real-time predictions.
3. **To design an intuitive and visually appealing interface** that enhances user experience, especially for medical professionals and researchers.
4. **To create a responsive system** that handles image preprocessing, model inference, and result display seamlessly.
5. **To evaluate the performance** of the classification model using appropriate metrics to ensure high accuracy and reliability.

3. Technology Stack and Tools Used

The HematoVision_AI project combines machine learning and web development technologies to deliver an end-to-end blood cell classification system. Below is a summary of the tools and technologies employed:

1. Programming Languages

- Python: Used for model development, image processing, and backend logic.
- HTML/CSS: Used for frontend design of the web interface.

2. Libraries and Frameworks

- TensorFlow / Keras: For building and training the Convolutional Neural Network (CNN) model.
- NumPy & Pillow (PIL): For image preprocessing and numerical operations.
- Flask: A lightweight Python web framework used to build the web application.
- Jinja2: For dynamic rendering of HTML templates in Flask.

3. Tools

- Google Colab: Utilized for model training due to its GPU support and ease of use.
- VS Code (Visual Studio Code): Used as the primary code editor for Flask development.
- Git & GitHub: Used for version control and to publish the project repository online.

Dataset Description

Source of Dataset

The dataset used in this project is the **Blood Cell Images Dataset** obtained from Kaggle – a publicly available dataset for classification of white blood cells. It is commonly used for research and machine learning tasks in hematology.

Number of Images :

The dataset contains a total of approximately **12,444 labeled images**, which are divided into training and test sets as follows:

- **Training Set:** 10,000+ images
- **Test Set:** 2,400+ images

Each image is labeled according to one of the four white blood cell types and is typically sized at **320x240 pixels**.

Classes :

The dataset includes four classes of white blood cells:

1. **Neutrophil**
2. **Eosinophil**
3. **Lymphocyte**
4. **Monocyte**

These categories are important in diagnosing various blood-related diseases and infections.

Training the Model:

Preprocessing Techniques

Before training the deep learning model, several preprocessing steps were applied to prepare the dataset:

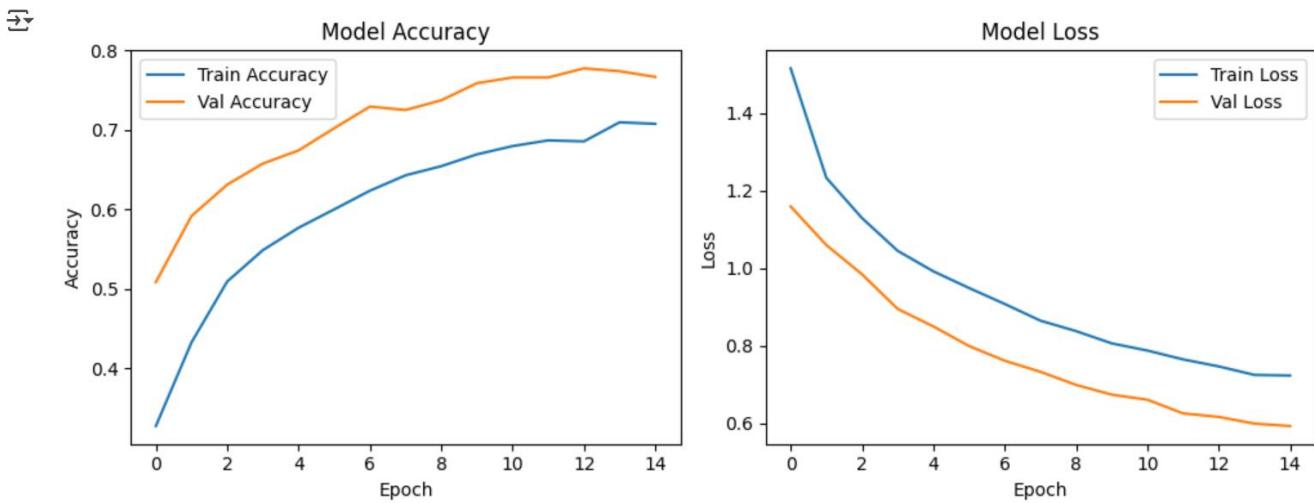
- Image Resizing: All images were resized to a fixed dimension of 224x224 pixels to match the input size expected by the pre-trained CNN model (e.g., MobileNetV2, ResNet, etc.).
- Normalization: Pixel values were normalized to a [0, 1] range by dividing by 255 to ensure faster convergence during training.
- Label Encoding: The class labels (Neutrophil, Eosinophil, Lymphocyte, Monocyte) were one-hot encoded.
- Data Augmentation: To reduce overfitting and enhance model generalization, image augmentation techniques were used:
 - Rotation
 - Horizontal/Vertical flipping
 - Zoom
 - Width and height shift

Dataset Splitting:

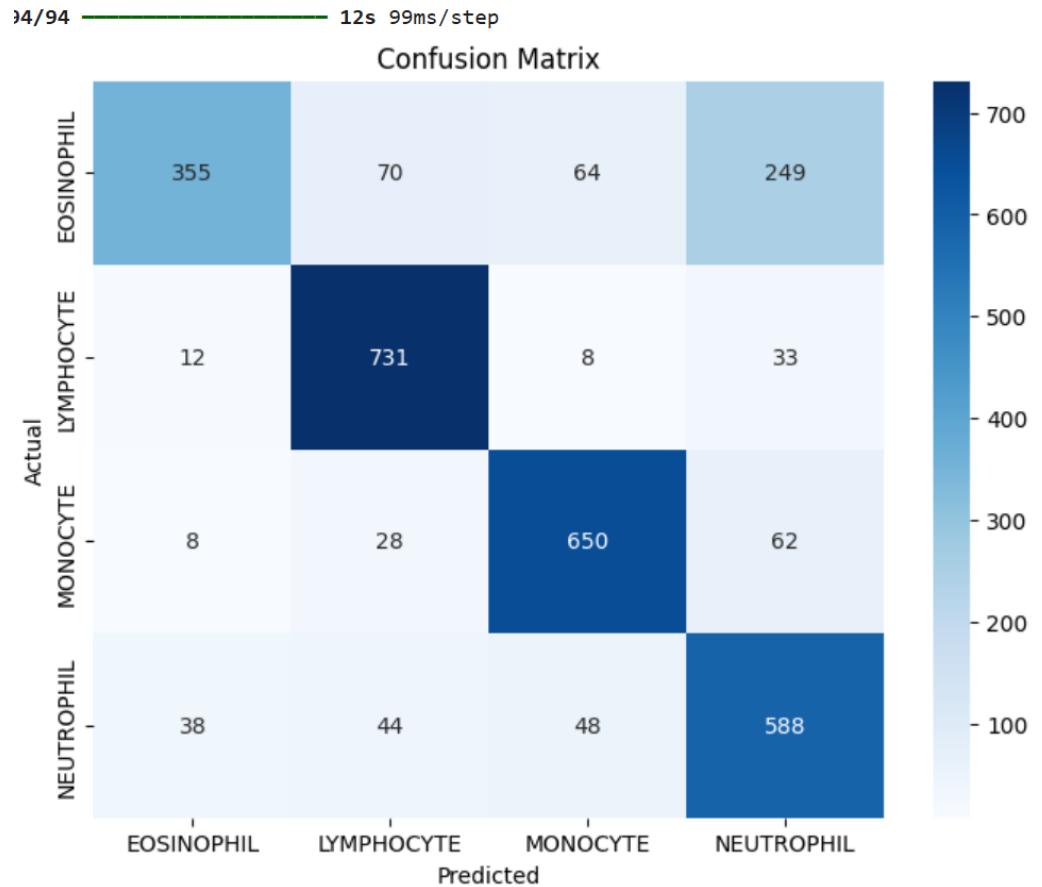
The dataset was split into three subsets:

- Training Set: 70% of the dataset
- Validation Set: 15% of the dataset
- Test Set: 15% of the dataset

Accuracy & Loss Graphs:



Confusion Matrix:



Flask Web Application:

To make the trained deep learning model accessible through a user-friendly interface, a Flask web application was developed. This app allows users to upload blood cell images and receive instant predictions.

HematoVision_AI/

```
|  
|   static/          # CSS, JS, Images  
|   |   css/  
|   |   uploads/     # Uploaded images  
|  
|   templates/      # HTML templates  
|   |   index.html    # Home page  
|   |   predict.html   # Upload & result page  
|  
|   model/  
|   |   BloodCell_model.h5  # Trained model  
|  
|   app.py          # Main Flask app  
|   requirements.txt # Dependencies  
└── README.md       # Project documentation
```

Home.html:

The screenshot shows a web browser window titled "Welcome to HematoVision". The address bar indicates the URL is "127.0.0.1:5000". The main content area features a red header bar with the text "Welcome to the HematoVision". Below this, a section titled "About Blood Cells" contains a paragraph about the importance of blood cells. A "Predict Blood Cell Type" section includes a file input field labeled "Choose File" with the placeholder "No file chosen" and a red "Predict" button.

Result.html:

The screenshot shows a web browser window titled "127.0.0.1:5000/predict". The main content area features a red header bar with the text "Prediction Result". Below this, a message "Predicted Class: EOSINOPHIL" is displayed above a thumbnail image of a blood smear showing a purple-stained eosinophil. At the bottom, there is a red "Upload Another Image" button.

Deployment Overview:

To make the HematoVision_AI blood cell classifier accessible to users without needing to interact with code, the entire model was deployed as a Flask web application. The backend handles image uploads and predictions, while the frontend provides an interactive user interface.

Model Saving:

After training and evaluating the model, it was saved in the .h5 format using Keras. This format preserves:

- The complete model architecture
- Trained weights
- Optimizer configuration
- Compilation setting

`model.save('blood_cell_model.h5')` - This .h5 file is later loaded into the Flask app to perform real-time predictions.

Run the Flask Application

`python app.py`

Open in Browser

Go to <http://127.0.0.1:5000/> in your browser to access the web interface.

GitHub Repository

The complete source code, trained model, templates, and documentation for the **HematoVision_AI** project are available in a public GitHub repository.

 **Repository Name:** HematoVision_AI

 **GitHub Link:** https://github.com/naga-rohith/HematoVision_AI

Conclusion

Summary of What Was Achieved

In this project, a deep learning-based web application, **HematoVision_AI**, was successfully developed to classify different types of white blood cells from microscopic images. The main accomplishments include:

- **Dataset Preparation:** Utilized a publicly available blood cell image dataset from Kaggle.
- **Model Development:** Trained a CNN model using transfer learning to classify four types of white blood cells: Neutrophils, Eosinophils, Lymphocytes, and Monocytes.
- **Web Deployment:** Integrated the model into a Flask web application that allows users to upload blood cell images and receive instant predictions.
- **User Interface:** Designed a simple and interactive UI for image upload and result display.
- **GitHub Integration:** Uploaded the full source code, model, and documentation to a public GitHub repository for open access and collaboration.

This solution can assist healthcare professionals or researchers in the initial screening of blood samples by automating white blood cell identification.

Thank You...